Application No.: 10/607,829

REMARKS/ARGUMENTS

In response to the Office Action dated April 27, 2005, Applicants hereby submit a legible copy of the Software Appendix, "Appendix A", corresponding to pages 23-26 of the specification filed June 27, 2003. Applicants hereby state that no new matter is presented with this document.

If the Examiner has any questions pertaining to this application or feels that a telephone conference would in any way expedite the prosecution of the application, please do not hesitate to call the undersigned at (408) 731-5000.

The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account 01-0431.

Respectfully submitted,

By__________ May 13, 2005
Leticia R. Block Reg. 50,167

Attachments – Appendix A (6 pages)

Customer No.: 22886
Legal Department
Affymetrix, Inc.
3380 Central Expressway
Santa Clara, CA 95051
Tel: 408/731-5000
Fax: 408/731-5392

APPENDIX A
SOFTWARE APPENDIX

```
( fullcal.awk )
( taxes input from a POLYcnsp CEL filc (115 .times. 130) ans )
( extracts ratio information for cvery block on the chip )
BEGIN(
ratpatcutoff = 1.2
pattoggle = `yes`
base (0) = "T"
base (1) = "G"
base (2) = "C"
base (3) = "A"
name(0.0) = 'WI-563'
hcx(0.0) = `TAGCC`
name(1.0) = `WI-567`
hex(1.0) = `TCAGAG`
name(2.0) = `WI-597`
hex(2.0) = `TGGATA`
name(3.0) = `WI-681`
hex(3.0) = `AACTAA`
name(4.0) = `WI-801`
hex(4.0) = `CTTGAG`
name(5.0) = `WI-802`
hex(5.0) = `CATCCT`
name(6.0) = `WI-1099`
hcx(6.0) = `CAGATA`
name(7.0) = `WI-1147`
hex(7.0) = `ACGAGC`
name(8.0) = `WI-1325`
hex(8.0) = `CTCTAC`
name(9.0) = `WI-1417`
hex(9.0) = `GTCTTT`
name(0.1) = `WI-1736`
hex(0.1) = `AAAGTC`
name(1.1) = `WI-1825`
hex(1.1) = `GTCTTC`
name(2.1) = `WI-1879`
hex(2.1) = `TACTCT`
name(3.1) = `WI-1888`
hex(3.1) = `ATGACA`
name(4.1) = `WI-1912`
hex(4.1) = `TTCTTT`
name(5.1) = `WI-1959`
hex(5.1) = `TCTCGG`
name(6.1) = `WI-1741`
hcx(6.1) = `GAAGGC`
name(7.1) = `WI-1760`
hex(7.1) = `ACCACA`
name(8.1) = `WI-1799`
hex(8.1) = `TCGATA`
name(9.1) = `WI-1973`
hex(9.1) = `CAAGAG`
name(0.2) = `WI-1980`
hex(0.2) = `AACTGA`
name(1.2) = `WI-2015`
hcx(1.2) = `GACTGT`
```

```
name(2.2) = `WI-2664`
hex(2.2) = `GGAGAG`
name(3.2) = `WI-4013`
hex(3.2) = `CTAGTG`
name(4.2) = `WI-7567`
hex(4.2) = `TAGTGA`
name(5.2) = `WI-11595`
hex(5.2) = `TAGAGC`
name(6.2) = `CM4.16`
hex(6.2) = `GATAAT`
name(7.2) = `WI-6704`
hex(7.2) = `ACTCCA`
name(8.2) = `WI-6731`
hex(8.2) = `GGCACA`
name(9.2) = `WI-6787`
hex(9.2) = `ACAGTT`
name(0.3) = `WI-6910`
hex(0.3) = `TAGTTG`
name(1.3) = `WI-9518`
hex(1.3) = `TTGATT`
name(2.3) = `ADM-3`
hex(2.3) = `ATAGTT`
name(3.3) = `AGT`
hex(3.3) = `TACTGG`
name(4.3) = `ALDOB-1`
hex(4.3) = `TTCTCG`
name(5.3) = `ALDOB-2`
hex(5.3) = `CCAGAT`
name(6.3) = `APO3`
hex(6.3) = `ACTCCT`
name(7.3) = `APOE(152T/C)`
hex(7.3) = `TGTCGC`
name(8.3) = `APOE(290T/C)`
hex(8.3) = `AGTCGC`
name(9.3) = `AR88`
hex(9.3) = `TCGATG`
name(0.4) = `AT1a`
hex(0.4) = `CTTCCC`
name(1.4) = `AT1b`
hex(1.4) = `GCACTT`
name(2.4) = `BCL2`
hex(2.4) = `ACGAGG`
name(3.4) = `BRCA1a`
hex(3.4) = `CATCTG`
name(4.4) = `ERCA1b`
hex(4.4) = `AGAGAG`
name(5.4) = `ERCA1c`
hex(5.4) = `GAAGAC`
name(6.4) = `D3S2`
hex(6.4) = `CCAGGT`
name(7.4) = `D3S11`
hex(7.4) = `TCTGAA`
name(8.4) = `D3S12`
hex(8.4) = `CCAGGG`
name(9.4) = `DRD2`
hex(9.4) = `CACTGG`
```

```
name(0.5) = `FABF2`
hex(0.5) = `GCGACT`
name(1.5) = `GCK`
hex(1.5) = `GAGACA`
name(2.5) = `NT2`
hex(2.5) = `CTGTGG`
name(3.5) = `HT2`
hex(3.5) = `TGCAAT`
name(4.5) = `HT4`
hex(4.5) = `ACTCGA`
name(5.5) = `HT5`
hex(5.5) = `GGGACC`
name(6.5) = `IGF2`
hex(6.5) = `TCTCGA`
name(7.5) = `IMS`
hex(7.5) = `TCTACC`
name(8.5) = `LDLA`
hex(8.5) = `GGCTAA`
name(9.5) = `LF79`
hex(9.5) = `CCAGGG`
name(0.6) = `LFL`
hex(0.6) = `AGCTAG`
name(1.6) = `NCC`
hex(1.6) = `GCCTGA`
name(2.6) = `METM`
hex(2.6) = `CCCTGG`
name(3.6) = `NEAMF`
hex(3.6) = `CAGATG`
name(4.6) = `FAR`
hex(4.6) = `ACATTG`
name(5.6) = `Per/RDS`
hex(5.6) = `GAAGGA`
name(6.6) = `PPP3R1`
hex(6.6) = `GACTAA`
name(7.6) = `RDS`
hex(7.6) = `AGGACG`
name(8.6) = `s14544`
hex(8.6) = `TCTGCT`
name(9.6) = `518CA`
hex(9.6) = `GGCATG`
name(0.7) = `TcA-CA1`
hex(0.7) = `TGCGGT`
name(1.7) = `TcR-CB22`
hex(1.7) = `GGCTGG`
name(2.7) = `TcR-CB23`
hex(2.7) = `CTCTAG`
name(3.7) = `TcR-CB24`
hex(3.7) = `GTGATG`
name(4.7) = `TcR-CB25`
hex(4.7) = `GTAGCC`
name(5.7) = `TcR-CB27`
hex(5.7) = `ACCTTA`
name(6.7) = `VB12a`
hex(6.7) = `ACAGTG`
name(7.7) = `VB12b`
hex(7.7) = `CACTCA`
```

```
bxgsum = 0
bxgnum = 0
readthis = 1
if (S1 ~ /(A-Za-z)/ ¦ S2 - /(A-Za-z1/) readthis = C
if (readthis == 1) rawdata/S1.S2) = S3
if (S1 > 2 && S2 > 4) if (S1 < 112 && S2 < 124) if (S1 < 90  .| S2 < 109)
  (
  px = int ((S1 - 3)/11)
  py = int ((S2 - 5)/15)
  pxo = (11*px) = 3
  pyo = (15*py) = 5
  mx = S1 - pxo
  by = S2 - pyo
  block = 3*(int(by/51) = 7
  if (by%5 |=  4 == mx |=  10)
     (
     sb = base(by%5)
     sig(px,py,block,SD,mx) = S3
     )
  if *by%5 == 4 || mx == 10)
     (
     bkgsum == S3
     bkgnum++
     )
  )
)
END(
printf ("background = %5.2f\n". bkgsum/bkgnum;
printf "MARKER\EBSTBLK\tRATIO\t\tDB\tCHECK\t\tPATRAT\n"
for (py = 0, py < 8, py++, for (px = 0, px < 10, px++) if (py < 7  || px < 8)
  (
  m(0) = substrihex(px,py),1.1)
  m(1) = substrihex(px,py),1.1)
  m(2) = substrihex(px,py),2.1)
  m(3) = substrihex(px,py),2.1)
  m(4) = substrihex(px,py),3.2)
  m(5) = substrihex(px,py),3.2)
  m(6) = substrihex(px,py),5.1)
  m(7) = substrihex(px,py),5.1)
  m(8) = substrihex(px,py),6.1)
  m(9) = substrihex(px,py),6.1)
  center = substrihex(px,py),3.1)*/*substrihex(px,py),4.1)
  pantmer = m(0)**m(2)*("center")*m(6)**m(8)
  header = "(`px + 1`, `py - 1`) " name(px,py)) "\n* pentamer *\n"
  headprint = 0
  (
     for (j = 0;j <= 2; j++ )
        (
        block = (3*j) + 7
        num2 = 0
        den2 = 0
        num1 = 0
        den1 = 0
        x2 = 0
        n1 = 0
        n2 = 0
```

```
for (f = 0; f < 5; f++)
  (
  maxhi (px,py,block,f) = 0
  for (g = 0; g < 4; g++) maxio(px,py,black,g,f) = 0
  )
for (k = 0; k <= 4; k++) for (b = 0; b <= 3; b++
  (
  z = int(k/2)
  signal = sig(px,py,block,bass(b),k)
  omit = 0
  if (mik) - bass(b)) omit = 1
  if (omit == 1)
    (
    q = maxhi (px,py,block,z)
    if (signal > q) maxni (px,py,block,b,z) = signal
      )
    if (omit == 0)
      (
      q = maxio (px,py,block,z)
      if (signal > q) maxio
        (px,py,block,b,z) = signal
      if (k42 == 0)
        (
        num2 == signal
        x2 == (signal) 2
        n1++
        )
      if (k42 == 1)
 (
 dan2 == signal
 x2 == (signal) 2
 n2++
 )
 )
   if (omit == 1) if (k == 4)/ k == 5,
 (
 if (base(b) == substr(hex|px,py), 3.1))
   (
   num1 == signal
   )
 if (base(b) == substr(hex|px,pay), 4.1))
   (
   dan1 == signal
   )
   )
 )
 maxhisum == maxhi(px,py,block,f)
 )
maxhisum = 0
for (f = 0; f < 5; f++)

  (
maxhisum = maxhisum/5
maxiosum = 0
for (g = 0; g < 5; g++)  for (v = 0; v < 4; v,g)
  (
        maxiosum += maxio(px,py,block,v,g)
        )
maxiosv = maxiosum/14
maxrat = maxniav/maxioav
num = ((num1/2) - (num2/n1))
if (num < 0) num = 0
dan == 0 ((dan1/2) - (dan2/n2))
if (dan < 0) dan = 0.001
ratio = num/dan
max = num1/2
if (dan1/2 < max l max = dan1/2
n = n1 + n2
stdvxnum = ((n*x2) - (num2 + dan2) ^2)
if (stdvxnum < 0) stdvx = 0
stdvx = (stdvxnum/(n ^2)) ^ (0.05)
if (maxrat > ratpatcutoff || pattoggle == `no`)
    (
    if (headprint == 0)
        (
        printf header
        headprint = 1
        )
    printf "\t20/"block"\t"
    printf ("%1.2f\t", ratio)
    if (ratio < 10000) printf "\t"
    rat = ratio
    if (ratio == 0) rat = .00001
    lograt = log(rat)/log(10)
    printf ("%2.2f\t", 10*lograt)
    printf ("%2.2f", max/stdvx)
    if (max/stdvx < 2) printf "\tFAIL\t"
    if (max/stdvx <= 2) printf "\t\t"
    printf ("%2.2f", maxrat)
    if (maxrat > ratpatcutoff) printf "\t*GOODFAT*"
    printf "\n"
    )
  )
)
```